Using Software Process Simulation to Assess the Impact of IV&V Activities[1]

David M. Raffo+*, Umanath Nayak*, Siri-on Setamanit,*
Patrick Sullivan*, Wayne Wakeland**

+College of Engineering and Computer Science
*School of Business Administration
**Systems Science Ph.D. Program

Portland State University
Portland, Oregon, USA

davidr@sba.pdx.edu, nayak@pdx.edu, sirion@pdx.edu, patrick@expatrick.com, wakeland@pdx.edu

Abstract

Today, organizations like NASA and the US Department of Defense make heavy use of Independent Verification and Validation (IV&V) techniques to improve the quality of systems and reduce the risks associated with the deployment of those systems. As more firms move to global outsourcing of their software development projects, IV&V will become increasingly valuable to companies as part of the their project activities. As a result, there is a critical need for cost effective IV&V. Software process simulation models (SPSMs) can be used to quantify the costs and benefits associated with both V&V and IV&V practices on software projects, enabling management to effectively allocate scarce resources for V&V and IV&V activities. The goals of this research are to quantitatively assess the economic benefit of performing IV&V on software development projects and to optimize that benefit across alternative IV&V plans. In this paper we present a model of one NASA project using the IEEE 12207 software development process with several possible IV&V configurations and assess their costs and benefits.

Keywords: Software Process Modeling, Software Process Simulation, Independent Verification and Validation, Software V&V, Software testing, Business Case Analysis

1 Introduction

Releasing defective software systems is costly, both in terms of the money and the risk involved, especially when human lives depend on the proper functioning of the systems. Detecting defects earlier in the development life cycle can lead to substantial reductions in rework effort and hence the duration of the project. Independent Verification and Validation (IV&V) can help to improve the quality and reliability of software systems, and thereby assure that the delivered products fully satisfy the user's operational needs.

Currently, organizations like NASA and the US Department of Defense make heavy use of IV&V techniques to improve the quality of systems and reduce the risks associated with the deployment of those systems. Moreover, the use of IV&V is increasing as firms move towards global outsourcing of software development. IV&V helps companies to assess the quality of the work they receive from off-shore contractors. Consequently, there is a critical need to assure that IV&V activities are effectively managed.

Software Process Simulation Models (SPSMs) can be used to quantify the costs and benefits associated with both V&V and IV&V practices on software projects, enabling management to more effectively allocate scarce resources for V&V and IV&V activities. In addition, SPSMs can help improve software development processes and product quality by assessing the performance of current V&V/IV&V deployments, and to evaluate new V&V/IV&V techniques and tools that might be applied.

The goal of the research presented in this paper is to quantitatively assess the economic benefit of performing IV&V on software development projects and to optimize that benefit across alternative IV&V plans. Our approach is based on extensive research into Software Process Simulation Models (SPSMs) conducted at the Software Engineering Institute (SEI) by Watts Humphrey, Marc Kellner, Bill Curtis, and others.

[1] This work has been sponsored by NASA Grant NAG5-12739

In this paper we present a model of one NASA project using the IEEE 12207 software development process and several possible IV&V configurations. We then assess the benefits and costs associated with these alternative configurations.

2 Background

2.1 What is Independent Verification and Validation (IV&V)?

Independent Verification and Validation methodology, as the name suggests, is performed by one or more groups that are completely independent of the developer of a system. IV&V techniques can be employed during all phases of the development life cycle with differing levels of coverage. Different IV&V techniques can also be employed in each phase with varying levels of defect detection capability and cost.

Some of the commonly used IV&V techniques that are listed in the IEEE 1012 standard which describes Verification and Validation (V&V) activities include [8 and 12]:

- Traceability Analysis
- Software Design Evaluation
- Interface Analysis
- Criticality Analysis
- Component Test Plan Verification
- Integration Test Plan Verification
- V&V Test Design Verification
- V&V Test Case Verification
- V&V Test Procedure Verification
- V&V Test Execution Verification
- Hazard Analysis
- Risk Analysis
- Source Code and Source Code Documentation Evaluation

When considering the use of IV&V on a project, the company must keep in mind that the use of IV&V processes may increase the budget and duration of the project. Furthermore, using every available IV&V technique may not make sense for every software project. A project might have a specific budget allocated for IV&V. So, while deciding the type of IV&V and different techniques to employ, project managers must carefully compare the various options available, and analyze the potential effectiveness of various IV&V techniques in order to best meet the project goals.

2.2 Could analytical software process models help to assess IV&V?

One approach for assessing the impact of process changes is the use of *cost estimation models* such as COCOMO [2], SLIM, [18] or other high-level models such as the work of Harrison and Eickelmann [5]. These models are limitied in that they do not capture the details associated with the development process or QA techniques being analyzed. As a result, these models cannot effectively assess variations of the IV&V techniques being applied to a particular project or process. For example, comparing the benefits of alternative insertion point options for a particular IV&V technique would be difficult when the process insertion points are not explicitly captured in the model.

Some past research sponsored by NASA has utilized various analytical models and techniques other than simulation to identify an optimal combination of IV&V techniques. However, most of the past work/models have examined the IV&V process in isolation without considering the implications of particular IV&V techniques on specific types of projects.

2.3 What do we mean by software process simulation modeling (SPSM)?

A variety of simulation approaches have been applied to software development activities [11], [16] and [17]. Discrete event simulation (DES) models of specific software processes have been reported in the literature: [4], [6], [18] and [20] among others. These models have been useful in predicting the cost and benefits associated with a number of different process changes and process variations. However, these models have not been built with the goal of application to IV&V projects.

The system dynamics (SD) paradigm (continuous system simulation) [1], [13], and [23] has also been used to represent portions of development and QA processes in the software development process. The SD models have the advantage of being able to effectively represent dynamic project concerns such as worker motivation and schedule pressure. However, these models assume that all work products flowing through the system are identical.

Other researchers have represented the software development process from the view of the developer using artificial intelligence based rules [15]. These models are focused on supporting various process enactment environments. However, the low level of detail captured by these representations, in our view, obscures the cost/ benefit performance picture.

Accordingly, we believe that DES models representing the software development process as distinct process steps, as would be found in a work break down structure, offer the best approach for comparing various process alternatives in terms of cost and benefits. Specifically, in order to evaluate different V&V and IV&V techniques in different combinations on different projects, our experience indicates that the discrete event paradigm using stochastic simulation models is most appropriate.

In previous work, Raffo et al. developed a number of Software Process Simulation Models (SPSMs) to predict the impact of various quality assurance techniques in terms of cost, quality, and schedule [13], [19-20]. [22] at a variety of organizations that develop commercial, government and military applications. This work has been based on extensive research into software process modeling conducted at the Software Engineering Institute (SEI) by Watts Humphrey, Marc Kellner, Bill Curtis, David Raffo and others [3], [7], [9] and [10]. Raffo's research focuses specifically on the costs, benefits and return on investment (ROI) associated with implementing testing and inspection processes in various combinations throughout the process lifecycle. The result is an economic justification and business case for process improvement efforts that managers can understand and use when setting budgets and trading off among multiple process improvement activities.

Moreover, due to the extensive sensitivity or "What if" analyses that can be done with simulation, SPSMs can not only be used to plan for the expected case, but also they can be used to assess the impact of changes in the development environment on IV&V (and V&V) techniques, as well as determine ways to improve IV&V and V&V applications.

Some of the questions that can be addressed using SPSMs in the context of IV&V include:

1. What would be the costs and benefits associated with implementing a given IV&V technique on a selected software project?
2. How would the given technique contribute to the development process and the assurance of quality?
3. How would this technique work in conjunction with other V&V or IV&V techniques?
4. When using this IV&V technique, should it be applied before or after testing?
5. How will the application of a particular IV&V technique earlier in the IV&V process affect the decision to apply a particular IV&V technique in later phases?
6. What would be the impact if a selected IV&V technique was applied to different portions of the process or applied multiple times?
7. Should one use Full In-Phase, Partial, Endgame, or Audit-Level IV&V?
8. What would be the impact of using this IV&V technique for different types of projects?
9. How does the criticality level of the overall system and individual components affect the selection of a particular IV&V technique?
10. How would employing a particular combination of IV&V techniques affect the development phase of the project, and by how much?
11. If we use a particular combination of IV&V techniques, will we be able to complete the project on time and within allocated budget?
12. If we use a particular combination of IV&V techniques, will we be able to match or beat the quality expectation for the system?
13. What combination of IV&V techniques is required to maximize the quality of the system?
14. How is the duration of the IV&V effort impacted by the overall staffing level for the project? How will this affect the total project duration?
15. What would be the impact on cost and schedule if staff were to be added to the IV&V tasks?
16. What if the complexity or defect profiles for a particular project were different than expected?
17. What would be the impact if selected V&V techniques are handled as IV&V services?
18. Is it possible to determine an "optimal" set of IV&V and/or V&V activities for a given project?

To be able to answer questions such as these would make a project manager's life considerably easier. Armed with this additional information,

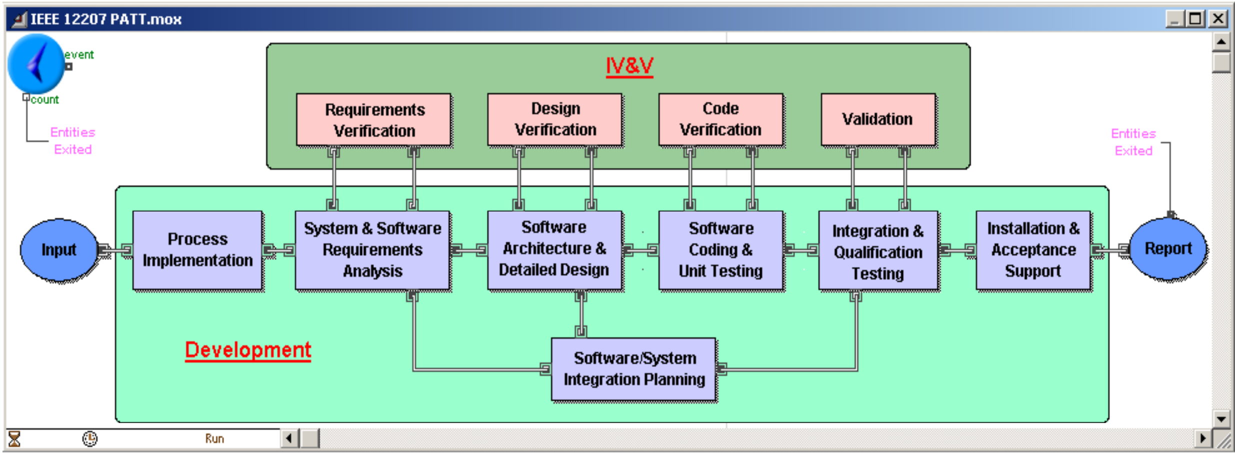

Figure 1: IEEE 12207 Software Process Base Model with IV&V Process

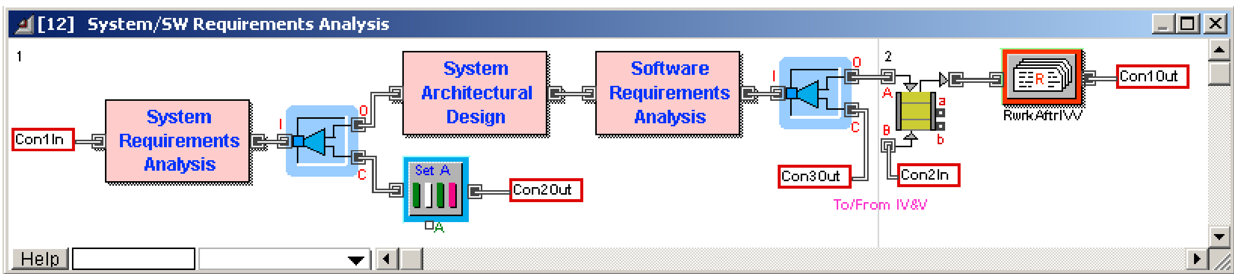

Figure 2: System/Software Requirements Analysis phase with Requirements Verification IV&V in sequence

he/she would be able to confidently decide what types of IV&V to employ and to identify the best combination of IV&V techniques to utilize on their particular project. This information would also enhance their ability to manage the inevitable tradeoffs between cost, quality, and project duration. Thus, building SPSMs to conduct cost/benefit analyses for IV&V techniques will make an important contribution to NASA mission success and assurance. We are currently working with NASA to do just that: to explore the use of software process simulation technology to model software development processes that incorporate IV&V methodologies and then use these models to predict the cost, quality, and schedule implications under a variety of conditions and situations.

3 Description of the Model

We have developed a prototype model of a NASA software project using a tailored IEEE 12207 software process as a base. The model was developed using Extend™ from ImagineThat, Inc. Data used in the model is a combination of data from NASA projects and industry standard data. *Figure 1* shows a high level screen shot from this model with IV&V phases added to it.

The process model is comprised of the following eight major lifecycle phases:

- Process Implementation
- System and Software Requirements Analysis
- Software Architecture and Detailed Design
- Software Coding and Unit Testing
- System Integration Planning
- Software Integration Planning
- Integration and Qualification Testing
- Installation and Acceptance Support

Each of these phases is a hierarchical block and has one or more process/activity steps in it. In total, there are 86 steps to the software development process. The IV&V portion of the model consists of Requirements Verification, Design Verification, Code Verification and Validation phases. *Figure 2* shows a configuration that implements the Requirements Verification IV&V sequentially after the completion of the System and Software Requirements Analysis phase.

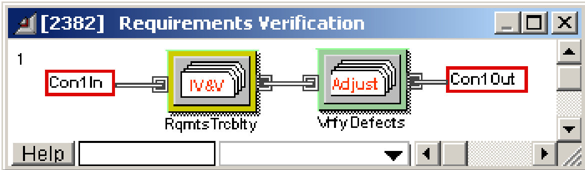

Figure 3: Requirements Traceability Technique Implementation

In the prototype model, we have implemented the Requirements Traceability IV&V technique in all four IV&V phases. *Figure 3* shows a screenshot of the Requirements Traceability technique implemented within the Requirements Verification phase. In addition, the model is able to enable or disable IV&V techniques based on the criticality level attribute of the artifacts that pass through them. This helps run the model with selected IV&V techniques in selected phases. Furthermore, we can add additional IV&V techniques either in parallel or in series after any IV&V technique.

As artifacts representing software code leave the development side and enter the IV&V activity, overall project effort and duration are affected as effort is spent in the IV&V phase. Modeling these IV&V activities in conjunction with the development activities will help the project manager gain an understanding of the overall effectiveness of employing particular IV&V techniques.

In the next section, we utilize the model to analyze a variety of IV&V implementations.

4 Preliminary Model Results

This section describes several use cases that illustrate the ability to use the model to quantitatively assess the economic and quality benefits of performing IV&V on software development projects, across alternative IV&V configurations.

To assess the benefit of performing IV&V, we will compare the baseline performance of the project (with no IV&V) with the project performance after the change has been implemented. We will first run the model to obtain baseline performance and then run the model with different configurations. Table 1 and table 2 show the results of the model (mean and standard deviation). Table 3 presents the percent improvement of each configuration compared to the baseline along each performance measure that is predicted by the current model.

4.1 Use Case 1: Assessing the Impact of IV&V at Different Points in the Development Process

What would be the impact of implementing Requirements Traceability Analysis at different development phases? How would this IV&V technique contribute to the overall performance of the software development process and the assurance of quality?

By applying Requirements Traceability Analysis late in the process at the Validation or Testing phase (Configuration 2), we are able to improve the quality of the product by reducing the latent defects by 8.8%. However, the cost and the duration of the project go up (by 2.6%), mainly due to the increase in rework since we detect more defects.

If we apply Requirements Traceability Analysis at Code Verification phase (Configuration 3), we are able to improve the product quality to approximately same level by reducing latent defects by 8.9%. In addition, the effort and duration also improve by 3.5% and 1.77% respectively (although the difference in duration is not statistically significant). Since we detect defects earlier (at Code vs. at Validation), the effort required to correct the defects will be less. As can be seen, the rework effort mean decreases by 21.2 person months when we employ Requirements Traceability Analysis at Code Verification instead of Validation phase. This reduction in rework effort contributes to the decrease in total project effort.

If we employ the IV&V technique at earlier phases (e.g., at Design or Requirements phase), the rework costs will be reduced even more. The rework effort mean decreases by about 9.1%-9.6% and the total effort mean reduces by 5.3%-5.6% for configurations 4 and 5. However, the defects that we pull out will be fewer since there will be fewer latent defects in the product at that point. As a result, we observe a 7.7% reduction in latent defects when applying Requirements Traceability IV&V at Design, and 4.7% when applying Requirements Traceability IV&V at Requirements.

4.2 Use Case 2: Assessing the Impact of Inserting Additional IV&V Techniques

What would be the impact of inserting an additional IV&V technique to work in conjunction

Case	Configuration	Total Effort Mean (Person Months)	Rework Effort Mean (Person Months)	Duration Mean (Months)	Corrected Defects Mean (Number of Defects)	Latent Defects Mean (Number of Defects)
1	Baseline	346.26	201.65	58.42	6,038.26	629.48
2	IV&V at Validation	355.35	210.75	59.95	6,113.79	574.17
3	IV&V at Code	334.13	189.53	57.38	6,134.84	573.49
4	IV&V at Design	327.93	183.33	56.56	6,123.11	581.27
5	IV&V at Requirements	326.82	182.21	56.40	6,078.87	600.04
6	IV&V at Code and Validation	342.14	197.54	58.78	6,203.66	524.96
7	IV&V at Req and Code	316.15	171.55	54.41	6,170.94	547.74
8	Two IV&V Techniques at Code	327.10	182.50	57.54	6,180.22	540.60
9	Dev Staff to 20	346.26	201.65	55.67	6,038.26	629.48
10	Dev Staff to 30	346.26	201.65	54.13	6,038.26	629.48
11	Dev Staff to 40	346.26	201.65	54.18	6,038.26	629.48
12	QA Staff to 20	346.26	201.65	58.42	6,038.26	629.48

Table 1: Results of the Model

	Total Effort Std. Dev. (Person Months)	Rework Effort Std. Dev. (Person Months)	Duration Std. Dev. (Months)	Corrected Defects Std. Dev. (Number of Defects)	Latent Defects Std. Dev. (Number of Defects)
Maximum	17.54	10.13	3.01	318.00	30.00
Minimum	9.86	5.85	1.66	169.00	15.00
Range	7.68	4.28	1.35	149.00	15.00

Table 2: Standard Deviation

	Configuration	Total Effort Mean	Rework Effort Mean	Duration Mean	Corrected Defects Mean	Latent Defects Mean
1	Baseline					
2	IV&V at Validation	-2.63%*	-4.51%*	-2.63%*	+1.25%	+8.79%*
3	IV&V at Code	+3.50%*	+6.01%*	+1.77%	+1.60%	+8.90%*
4	IV&V at Design	+5.29%*	+9.09%*	+3.17%*	+1.41%	+7.66%*
5	IV&V at Requirements	+5.62%*	+9.64%*	+3.46%*	+0.67%	+4.68%*
6	IV&V at Code and Validation	+1.19%	+2.04%	-0.63%	+2.74%	+16.60%*
7	IV&V at Req and Code	+8.69%*	+14.93%*	+6.86%*	+2.20%	+12.99%*
8	Two IV&V Techniques at Code	+5.53%*	+9.50%*	+1.50%	+2.35%	+14.12%*
9	Dev Staff to 20	0.00%	0.00%	+4.70%*	0.00%	0.00%
10	Dev Staff to 30	0.00%	0.00%	+7.33%*	0.00%	0.00%
11	Dev Staff to 40	0.00%	0.00%	+7.25%*	0.00%	0.00%
12	QA Staff to 20	0.00%	0.00%	0.00%	0.00%	0.00%

Table 3: Percent Improvement Compared to the Baseline

Note:
For effort, duration, and latent defects, "+" indicates less effort, shorter duration, or fewer latent defects.
For corrected defect, "+" indicates more defects corrected.
* significant at 95% confidence level

with the current IV&V technique? What combination of IV&V techniques provides the maximum improvement to project performance?

When implementing Requirements Traceability Analysis at Code Verification and a different IV&V technique such as Model Checking at the Validation phase in combination (configuration 6), there is a major improvement in the quality of the product. The latent defect number significantly decreases by 16.6%, while the effort and duration are not statistically significant different from the baseline.

On the other hand, if we implement Requirements Traceability Analysis at Requirements and also Model Checking at Code Verification phase (case 7), the latent defects will decrease by 13%. We do not pull out as many defects as configuration 6 since we apply the IV&V technique at an earlier phase and fewer defects are currently in the code. However, detecting defect earlier contributes to lower rework effort. As a result, we are able to improve the cost and duration compared to the baseline. The total effort decreases by 8.7% and the duration decreases by 6.7%.

In a third configuration, we apply both IV&V techniques (e.g., Requirements Traceability and Model Checking) at the same phase - at Code Verification phase (configuration 8). By doing this, we observe a higher quality product (latent defects reduce by 14.1%) while achieving improvement in cost and duration (total effort decreases by 5.5% and duration decrease by 1.5%). The reduction in effort is not as great as case 7 since we detect defects in later phase. However, we are able to pull out more defects.

It is important to note that examining and evaluating how IV&V techniques interact with each other is critical to this analysis. What would be the effectiveness of particular IV&V technique when it is the only technique used or when it is used in conjunction with other IV&V techniques? The effectiveness of an IV&V (or any V&V) technique changes depending upon the history of V&V/IV&V that has been applied previously to a particular work product. We believe that further research on this issue is needed. At the same time, useful information can be obtained from approximate estimates of this effect using our model and conducting "what if" analyses.

This preliminary analysis is presented only to illustrate the possible uses of our model. In order to determine an "optimal" set of IV&V and/or V&V activities for a given project, full business analysis would be needed.

When developing a business case, there are several ways to evaluate alternatives. Four approaches are suggested by Raffo and Kellner [21].

1. Simple comparison of performance measure deltas
2. Comparison of performance measure deltas using a utility function
3. Comparison of performance measure deltas using financial measures such as Net Present Value (NPV) and Return on Investment (ROI)
4. Comparison of overall performance measure value using Data Envelopment Analysis (DEA)

In order to conduct a full business case analysis (e.g., in order to compute NPV or ROI), we also need to include the cost of implementing and conducing IV&V. Table 4 shows the cost of IV&V.

In addition to using the model to assess the economic and quality benefits of performing IV&V on software development projects, the model can also be used to address human resource issues.

	Configuration	**IV&V Cost (Hour)**
2	IV&V at Validation	284.73
3	IV&V at Code	230.02
4	IV&V at Design	211.68
5	IV&V at Requirements	137.90
6	IV&V at Code and Validation	512.13
7	IV&V at Req and Code	362.78
8	Two IV&V Techniques at Code	404.26

Table 4: Average Cost of IV&V Effort (in hours) by Configuration

4.3 Use Case 3: Assessing the Impact of Adding Staff

What would be the impact on cost, quality, and schedule of the project if staff were to be increased or reduced?

By observing the resource utilization in the baseline model, we found that the current number of staff (16 Development staff and 15 QA staff) was not optimal. The model showed that some development work requests had to wait. When we increased the number of development staff to 20 (configuration 9), the duration reduced by 4.7%. Effort, corrected defects, and latent defects stayed the same.

With 20 development staff, there were still some work requests that were not fulfilled. Therefore, we increased development staff to 30 (configuration 10). We found that the duration further decreased (7.3%) with no change in other measures. However, when we increased development staff to 40 (configuration 11), there was no further improvement in the duration. The duration decreases from baseline at approximately the same as when we have 30 development staff. This is explained by the fact that there were no remaining unfulfilled requests for development staff.

We also increased QA staff size to 20 (configuration 12) to see whether it would have any effect on project performance measures. No impact was found. The current number of QA staff (15) was already sufficient. One indicator is that there is no request for QA staff that is not fulfilled in the baseline.

In order to properly support a decision regarding the number of staff on a project, more information would be required. The model assumes that all staff can be utilized productively on other projects when they are not actively working on the target project (i.e., that the cost of non-productive time is 0). This is unlikely to be the case. As a result, the cost of the project would increase because the project would also have to pay for the non-productive time of all staff members. This logic will be incorporated into a future version of the model.

The preceding use cases illustrate how project managers might use the model to help make decisions about IV&V activities, project resources and other issues. Armed with this information, the project manager will be in a better position to effectively allocate scarce resources to various V&V/IV&V activities. Note that these experiments are based on preliminary high-level data from actual NASA projects. We are in the process of collecting more detailed data and anticipate finer grained results in the future.

5 Conclusions

Today, organizations like NASA and the US Department of Defense make heavy use of IV&V techniques to improve the quality of systems and to reduce the risks associated with the deployment of those systems. As firms move towards global outsourcing of software development, the use of IV&V is increasing. IV&V helps companies to assess the quality of the work they receive from off-shore contractors. Consequently, there is a critical need to effectively manage IV&V.

Given proper data, software process simulation models (SPSMs) can be used to quantify the costs and benefits associated with both V&V and IV&V practices on software projects, thereby enabling management to more effectively allocate scarce resources for V&V and IV&V activities.

In this paper, we have presented a large-scale software development process model that is being developed and used at NASA. This model evaluated the effectiveness of applying IV&V at various points in the development process. We also presented preliminary illustrative results that show the types of analyses that can be done and pointed out some of the research issues associated with this work.

This research contributes to mission assurance and success by making recommendations as to how IV&V technologies should be deployed across various projects. These recommendations support planning and management of IV&V and enable IV&V technologies to be applied to software projects more quickly, in order to achieve greater benefits at lower cost.

6 References

[1] Abdel-Hamid, T. and Madnick, S., *Software Project Dynamics: An Integrated Approach*, Prentice-Hall Software Series, Englewood Cliffs, New Jersey, 1991

[2] Boehm, B. *Software Engineering Economics*, Prentice-Hall. Englewood Cliffs, New Jersey,, 1981.

[3] Curtis, B., Kellner, M. I., Over, J., "Process Modeling", *Communications of the ACM*, Vol. 35, No. 9, September, 1992.

[4] Donzelli and Iazeolla, "Hybrid Simulation Modelling of the Software Process", *Journal of Systems and Software*, Volume 59, Number 3, December 2001

[5] Harrison and Eickelmann, "Risk-Based Financial Analysis Tools and Techniques to Aid IV&V Decision Making", *Proceedings of the Software Engineering Research Center (SERC) Showcase*, Held at West Virginia University, May 28-30, 2002.

[6] Höst, Regnell, Dag, Nedstam, and Nyberg , "Exploring Bottlenecks in Market-Driven Requirements Management Processes with Discrete Event Simulation", *Journal of Systems and Software*, Volume 59, Number 3, December 2001

[7] Humphrey, W. and Kellner M. I., "Software Process Modeling: Principles of Entity Process Models", *Proceedings of the 11th International Conference on Software Engineering*, IEEE, May 1989, pp. 331-342.

[8] *IEEE Standard for Verification and Validation*, IEEE Std 1012, IEEE Computer Society, New York, July 1998.

[9] Kellner, M., "Software Process Modeling Experience", *Proceedings of the 11th International Conference on Software Engineering* (Held at Pittsburgh, Pennsylvania, USA, May 15 - 18, 1989), IEEE, 1989, pp 400-401.

[10] Kellner, M.I., and Hansen, G.A., Software Process Modeling. Technical Report. CMU/SEI-88-TR-9, DTIC: ADA197137, Software Engineering Institute, Carnegie Mellon University, May 1988.

[11] Kellner, Madachy, and Raffo, “Software Process Modeling and Simulation: Why, What, How,” *Journal of Systems and Software*, Vol. 46, No. 2/3 (15 April 1999), pages 91-105.

[12] Lewis ,R. O., *Independent Verification and Validation: A Lifecycle Engineering Process for Quality Software*, John Wiley and Sons, New York, 1992.

[13] Madachy, R.J., *A Software Project Dynamics Model for Process Cost, Schedule, and Risk Assessment,* Ph.D. Thesis, University of Southern California, 1994.

[14] Martin, R.H. and D. M, Raffo, “Application of a Hybrid Process Simulation Model to a Software Development Project”, *Journal of Systems and Software*, Volume 59, Number 3, December 2001.

[15] Mi, P, and Scacchi, W., "Modeling Articulation Work in Software Engineering Process", *Proceedings of the First International Conference on the Software Process*, IEEE Computer Society, Washington, DC., 1991, pp. 188-201.

[16] *Proceedings of the International Workshop on Software Process Simulation Modeling (ProSim)*, Held in Silver Falls, Oregon, June 22-25, 1999.

[17] *Proceedings of the International Workshop on Software Process Simulation Modeling (ProSim)*, Held Imperial College, London, UK, July 10-12, 2000.

[18] Putnam, L. H., "General Empirical Solution to the Macro Software Sizing and Estimating Problem", *IEEE Transaction on Software Engineering*, SE 4, 4, July 1978 pp. 345-361

[19] Raffo, David M., “*Modeling Software Processes Quantitatively and Assessing the Impact of Potential Process Changes on Process Performance*”, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1996. Director: Dr. Marc I. Kellner, *Software Engineering Institute (SEI)*

[20] David M. Raffo and Marc I. Kellner, “Predicting the Impact of Potential Process Changes: A Quantitative Approach to Process Modeling,” *Elements of Software Process Assessment and Improvement*, IEEE Computer Society Press, 1999

[21] David M. Raffo and Marc I. Kellner, “Empirical Analysis in Software Process Simulation Modeling,” *Journal of Systems and Software*, Vol. 47, No. 9 (2000).

[22] Raffo, Vandeville, and Martin, “Software Process Simulation to Achieve Higher CMM Levels,” *Journal of Systems and Software*, Vol. 46, No. 2/3 (15 April 1999), pages 163-172.

[23] Tvedt, J., A System Dynamics Model of the Software Inspection Process, Technical Report TR-95-007, Arizona State University, Tempe, Arizona, 1995.